Diplomski studij

Informacijska i komunikacijska tehnologija

Telekomunikacije i informatika

Računarstvo

Programsko inženjerstvo i informacijski sustavi

Računalno inženjerstvo
Računarska znanost

Ak. g. 2016./2017.

Raspodijeljena obrada velikih skupova podataka

2. Domaća zadaća

1. Zadatak: Izrada MapReduce programa

Cilj zadatka je uspješno napisati, prevesti te izvršiti MapReduce program u pseudo-raspodijeljenoj inačici platforme Hadoop (tj. izvršiti ga unutar virtualnog stroja).

Uspješnim rješenjem ovog zadatka steći ćete sljedeća znanja:

- **osnove rada s postavkama programa prilagođenog za programski model MapReduce**
- **definiranje funkcija Map i Reduce uz korištenje složenih struktura podataka**

> **IZVJEŠTAJ:** Na sustav Moodle predajete izvorni programski kôd te odgovore na pitanja koja slijede iza opisa zadataka.

Za potrebe ove zadaće pripremljena je datoteka `trip_data1.csv` s natjecanja DEBS Grand Challenge 2015 koja sadrži informacije o vožnjama taksija na širem području grada New Yorka. Datoteku možete dohvatiti sa sljedeće poveznice (http://svn.tel.fer.hr/trip_data.zip) te proučite njezin sadržaj. Nakon toga ju prenesite na HDFS u mapu po izboru.

Detaljni opis zadatka:

U ovom zadatku ćete napraviti program MapReduce koji će obraditi podatke iz ulazne datoteke tako da odredite za svako taksi vozilo koje se pojavljuje u ulaznoj datoteci **ukupno trajanje vožnje u sekundama** koje je taksi vozilo prešlo za sve vožnje, trajanje najkraće vožnje i trajanje najduže vožnje za svako vozilo.

Funkcija Map treba proći kroz ulaznu datoteku, za svaku vožnju identificirati i izdvojiti dva potrebna parametra (oznaka taksija – *`medallion`* i duljina vožnje – *`trip_time_in_secs`*). Funkcija Reduce treba zatim za svako vozilo odrediti ukupno trajanje puta koje je vozilo ostvarilo na temelju svih podataka iz datoteke `trip_data1.csv` te maksimalno odnosno minimalno trajanje pojedine taksi vožnje za svaki taksi. Sva tri tražena parametra potrebno je izračunati tijekom izvođenja samo jednog MapReduce posla. Osim funkcija Map i Reduce, potrebno je definirati „*driver*“ za MapReduce posao (tj. konfigurirati, pokrenuti i izvršiti MapReduce Job) tako da konfigurirate i instancirate inačicu posla, postavite putanju do ulazne i izlazne datoteke, postavite funkcije Map i Reduce te pričekate kraj izvršavanja posla unutar Hadoop okoline.

Nakon što ste uspješno izvršili prvi zadatak, promijenite konfiguraciju MapReduce programa (a po potrebi i izvorni kod pojedinih klasa) tako da vaš posao koristi funkciju Combiner. Napomena: zabilježite vrijeme izvođenja zadatka sa i bez Combinera.

Nakon uspješnog pokretanja oba programa odgovorite na sljedeća pitanja:

- Koliko različitih vozila se nalazi u ulaznoj datoteci?
- Koliko je trajala najdulja ukupna vožnja jednog taksija? Koja je minimalna, a koja najdulja vožnja tog taksija?
- Koje ste promjene morali napraviti na izvornom kodu prilikom uvođenja optimizacijske funkcije Combine?
- Koliko vremena su se izvodile inačice programa? Je li to u skladu s vašim očekivanjem? Objasnite zašto.

2. Zadatak: Korištenje obrasca dijeljenja podataka (*partitioning pattern*)

Cilj zadatka je napisati MapReduce program koji implementira obrazac dijeljenja podataka prema definiranom kriteriju (*partitioning pattern*)

Uspješnim rješenjem ovog zadatka steći ćete sljedeća znanja:

- **oblikovanje MapReduce programa prema zadanom obrascu**
- **prilagodbu postavki MapReduce programa (korištenje partitionera)**

> **IZVJEŠTAJ:** Na sustav Moodle predajete izvorni programski kôd te odgovore na pitanja koja slijede iza opisa zadataka.

Detaljni opis zadatka:

Proučite obrazac dijeljenja podataka prema definiranom kriteriju te osmislite generički pseudo-kod koji ostvaruje zadanu funkcionalnost. Zatim osmislite rješenje koje će omogućavati da se taksi vožnje iz ulaznog skupa podataka podijele u šest skupina. Osnovna podjela je po području vožnje na vožnje koje imaju početak i kraj u užem centru New Yorka i vožnje koje imaju početak ili kraj izvan užeg centra grada. Vožnja pripada užem centru grada ako započne i završi unutar područja omeđenog koordinatama (longitude, latitude) [-74, 40.8] i [-73.95,40.75]. Dodatno podijelite vožnje iz prethodne dvije skupine i prema broju putnika tako da stvorite tri skupine: 1 putnik, 2-3 putnika te 4 i više putnika. Program treba napraviti izlazne datoteke za svaku spomenutu skupinu koje će sadržavati podatke o vožnji istovjetne ulaznom podatkovnom skupu.

Funkcija Map treba proći kroz ulaznu datoteku, za svaku vožnju identificirati i izdvojiti potrebne parametre za podjelu vožnji prema jednom od definiranih kriterija. Funkcija Partition treba izlazne podatke iz Map funkcije podijeliti prema dodatnom kriteriju u manje skupove podataka. Funkcija Reduce treba zapisati dobivene podatke u jednu datoteku čime će svaka skupina biti obrađena jednom Reduce funkcijom (na kraju trebate dobiti 6 izlaznih datoteka). Osim funkcija Map, Partition i Reduce potrebno je definirati „driver“ za MapReduce program (tj. MapReduce Job) tako da konfigurirate i instancirate inačicu posla, postavite putanju do ulazne i izlazne datoteke, postavite funkcije Map, Partition i Reduce, postavite broj Reduce poslova na odgovarajući broj te pričekate kraj izvršavanja posla unutar Hadoop okoline.

Nakon uspješnog pokretanja odgovorite na sljedeća pitanja:

- Koliko je vožnji realizirano u pojedinom području, tj. u užem centru i u širem gradskom području, i to tako da je broj putnika bio 1, 2-3 putnika ili 4 i više putnika?
- Koje ste promjene morali napraviti na izvornom kodu prilikom uvođenja funkcije Partition?
- Koliko je vožnji navedeno u svakoj podskupini?

3. Zadatak: Izvršavanje ulančanih MapReduce programa

Cilj zadatka je napisati program koji se sastoji od dva ulančana MapReduce programa. Prvi MapReduce program priprema podatke za obradu, dok drugi MapReduce program obrađuje podatke i zapisuje ih u izlaznu datoteku.

Uspješnim rješenjem ovog zadatka steći ćete sljedeća znanja:

- **definiranje ulančanog MapReduce posla**

IZVJEŠTAJ: Na sustav Moodle predajete izvorni programski kôd i odgovore na pitanja.

Detaljni opis zadatka:

Ovaj zadatak se temelji na prethodna dva, tj. cilj je koristiti novi MapReduce program koji ulančava prethodna dva zadatka. Program treba za svako vozilo identificirati ukupno trajanje svih vožnji te minimalno i maksimalno trajanje jedne taksi vožnje ovisno o tome je li vožnja ostvarena u užem centru New Yorka ili u širem gradskom području i to posebno prema broju putnika (1 putnik, 2-3 putnika te 4 i više putnika).

Potrebno je definirati prvi MapReduce posao koji će podijeliti podatke u šest skupova (2. zadatak), a na temelju kojih će drugi MapReduce program izvršiti obradu podataka tih podskupova podataka kako bi za svaki podskup izračunao ukupno trajanje vožnji, minimalnu i maksimalnu vožnju za svaki taksi. Posao možete ulančati u programskom kodu „drivera“ pomoću metode Job.waitForCompletion. Pazite na tijek izvođenja poslova, provjerite uspješnost izvršenja prvog definiranog posla prije nego što počne izvršavanje drugog itd. Također pravilno definirajte ulazne podatke za novi posao. Na kraju programa obrišite međurezultate (izlaz iz svakog MapReduce posla koji nije posljednji) programski pomoću naredbe:

```
FileSystem.get(conf).delete(TEMP_RESULT, true);
```

Nakon uspješnog pokretanja odgovorite na sljedeće pitanje:

- Koliko ste MapReduce poslova izvršili u vašem kôdu?

Koliko je različitih taksija realiziralo vožnje u pojedinom području, tj. u užem centru i u širem gradskom području, i to tako da je broj putnika bio 1, 2-3 putnika ili 4 i više putnika?